

REMARKS

Applicant thanks Examiner McLeod and Examiner Jacobs for their time and attention, as well as their helpful suggestions regarding this matter, during the January 26, 2009, Examiner's Interview with Applicant's undersigned attorney. Applicant further acknowledges, with thanks, receipt of the Interview Summary. As indicated in the Summary, during the interview the Traversat and Chen references and possible proposed claim amendments were discussed. No agreement regarding the claims was reached.

§112 Rejections

The Examiner rejected claims 1, 5, 9, 10 and 11 under 35 U.S.C. §112, first paragraph on the ground that there is no support in the specification for the language "a business application to manage and control a business enterprise". The examiner also rejected claims 1, 5, 9, 10 and 11 under 35 U.S.C. §112, second paragraph on the ground that it is not clearly understood how or what the business application does or is doing to manage and control a business enterprise.

To expedite prosecution of the above-identified application, applicant amended independent claim 1 to replace the recitation "a business application to manage and control a business enterprise" with the recitation "a business application of a collaborative business enterprise". Support for the substituted recitation is provided throughout the application, including, for example, in the title and abstract, at page 1, paragraphs 6-11, page 2, paragraph 24, etc., of the published application (US 2005/0226240). Applicant similarly amended independent claims 5, 9, 10 and 11.

102 and 103 Rejections

The examiner rejected claim 12 under 35 U.S.C. § 102(b) as being anticipated by U.S. Publication No. US2002/0184357 to Traversat *et al* (in that regard, applicant notes that claim 12 depends from independent claim 11 which the examiner rejected under 35 U.S.C. § 103(a);

applicant assumes that this rejection was made in error). Additionally, the examiner rejected claims 1, 5, 9-11 and 13 under 35 U.S.C. § 103(a) as being unpatentable over Traversat in view of U.S. Publication No. US2002/0184070 to Chen, rejected claims 2-4 and 14-15 under 35 U.S.C. § 103(a) as being unpatentable over Traversat in view of Chen, and further in view of U.S. Publication No. US2003/0014733 to Ringseth *et al.*, and rejected claims 6-8 under 35 U.S.C. § 103(a) as being unpatentable over Traversat in view of Chen, further in view of Ringseth, and further in view of U.S. Publication No. 20020138618 to Szabo. These rejections are respectfully traversed.

Applicant amended independent claim 1 to clarify that the content and configuration of the message header is based, at least in part, on a message class of the application message. Applicant further amended independent claim 1 to recite that the message class can include one of multiple possible values including, for example, a first value representative of an application-message class associated with application messages that cause specified operations to be performed at the receiving application, a second value representative of an application-response class associated with messages responsive to the application messages of the application-message class, a third value representative of an application-error class associated with error messages indicative of errors occurring at the receiving application processing the application messages and a fourth value representative of a system-acknowledge class associated with acknowledgement messages indicative that one or more application messages have been received by the receiving application. Support for this clarification is provided throughout the originally filed application, including, for example, in FIG. 3, and at pages 4-5, paragraphs 51-54 of the published application (PG US Patent Publication No. 2005/0226240). Applicant similarly amended independent claims 5, 9, 10 and 11.

Additionally, applicant amended independent claim 1 to clarify that the processing mode has one of a multiple of values indicative of whether a reply responsive to the application message is to

be transmitted to the sending application upon processing of the application message by the receiving application. Support for this amendment is provided, for example, at paragraph 34 of the published application. Applicant further amended independent claim 1 to recite that the message header may further include, a modifiable hop-list to record the identity of the intermediate components through which the application message passes en route to the receiving application. Support for this amendment is provided, for example, at paragraph 30 of the published application.

Applicant further added new claim 16, depending from claim 1, reciting the feature that the application-response class is associated with messages comprising return values responsive to respective computations performed by the receiving application in response to requests in the application messages received from the sending application. Support for these features is provided, for example, in paragraph 51 of the published application.

Applicant's independent claim 1 thus recites "defining an application message having a structured application message header, the structured message header being defined in accordance with a message class of the application message determinative of content and configuration of the message header and in accordance with a messaging protocol of a business application of a collaborative business enterprise." Thus, the particular content and configuration of the header of the application message being sent is based, at least in part, on the type of message class corresponding to the message. The header of the application message is also generated in accordance with a business application protocol. As explained in the above-identified application:

[0051] According to the messaging protocol, any of a number of message classes may be used to define derivative invariants of the messaging protocol. Each message class may be defined in view of the main purpose of a message. For example, messages may be defined in accordance with the classes ApplicationMessage, ApplicationResponse, ApplicationError, SystemAck, ApplicationAck, or SystemError. For those classes, ApplicationMessage may be a message that is sent to an application, ApplicationResponse may be a message that synchronously responds to an ApplicationMessage (e.g. a return value in response to an ApplicationMessage that requests a calculation by a component), ApplicationError may be a message that includes an error in response to an ApplicationMessage that was caused by an

application program, SystemAck may be a message acknowledging that a message has been received by a component of the system that implements the messaging protocol, ApplicationAck may be a message that informs the sender of the message about a successful or erroneous application execution of an ApplicationMessage at the final recipient, and SystemError may be a message that includes an error indicating that a system component generated an error. (2005/0226240, pages 4-5, paragraph 51)

As claim 1 further recites, the various message classes include the application-message class, the application-response class, the application-error class and the system-acknowledge class. Paragraphs 52-54 and FIG. 3 more particularly describe how the message header's content and configuration is affected by the particular message class of the application message.

In contrast, none of the references cited by the examiner discloses or suggests at least the features “defining an application message having a structured application message header, the structured message header being defined in accordance with a message class of the application message determinative of content and configuration of the message header,” as required by applicant’s independent claim 1. The cited references certainly do not disclose the various message classes recited in claim 1, where “the message class having one of multiple possible values including: a first value representative of an application-message class associated with application messages that cause specified operations to be performed at the receiving application, a second value representative of an application-response class associated with messages responsive to the application messages of the application-message class, a third value representative of an application-error class associated with error messages indicative of errors occurring at the receiving application processing the application messages and a fourth value representative of a system-acknowledge class associated with acknowledgement messages indicative that one or more application messages have been received by the receiving application.”

Specifically, Traversat describes peer-to-peer network computing platforms (Traversat, page 1, paragraph 7). Traversat explains that messages may be datagrams that may include an envelope

with a header, a message digest, (optionally) the source endpoint, and the destination endpoint, and further describes that in one embodiment each protocol header may include a tag naming the protocol in use and a body length:

[0147] In one embodiment, the peer-to-peer platform may use asynchronous messages as a basis for providing Internet-scalable peer-to-peer communication. The information transmitted using pipes may be packaged as messages. Messages define an envelope to transfer any kinds of data. A message may contain an arbitrary number of named subsections which can hold any form of data. In one embodiment, the messages may be in a markup language. In one embodiment, the markup language is XML. Each peer's messaging layer may deliver an ordered sequence of bytes from the peer to another peer. The messaging layer may send information as a sequence of bytes in one atomic message unit. In one embodiment, messages may be sent between peer endpoints. In one embodiment, an endpoint may be defined as a logical destination (e.g. embodied as a URN) on any networking transport capable of sending and receiving Datagram-style messages. Endpoints are typically mapped into physical addresses by the messaging layer at runtime.

[0148] In one embodiment, a message may be a Datagram that may include an envelope and a stack of protocol headers with bodies and an optional trailer. The envelope may include, but is not limited to, a header, a message digest, (optionally) the source endpoint, and the destination endpoint. In one embodiment, each protocol header may include, but is not limited to, a tag naming the protocol in use and a body length. Each protocol body may be a variable length amount of bytes that is protocol tag dependent. Each protocol body may include, but is not limited to, one or more credentials used to identify the sender to the receiver. Such a message format preferably supports multiple transport standards. An optional trailer may include traces and accounting information. (Traversat, page 12, paragraphs 147-148)

But nowhere does Traversat describe that the message header is defined in accordance with a message class of the application message that is determinative of the content and configuration of the message header. Traversat certainly does not describe that such message classes include classes such as the application-message class, the application-response class, the application-error class and/or the system-acknowledge class recited in applicant's claim 1. Accordingly, Traversat fails to disclose or suggest at least the features of "defining an application message having a structured application message header, the structured message header being defined in accordance with a message class of the application message determinative of content and configuration of the message header," required by applicant's independent claim 1 or "the message class having one of multiple

possible values including: a first value representative of an application-message class associated with application messages that cause specified operations to be performed at the receiving application, a second value representative of an application-response class associated with messages responsive to the application messages of the application-message class, a third value representative of an application-error class associated with error messages indicative of errors occurring at the receiving application processing the application messages and a fourth value representative of a system-acknowledge class associated with acknowledgement messages indicative that one or more application messages have been received by the receiving application,” required by applicant’s independent claim 1.

Chen describes a peer-to-peer collaborative process management method and system for supporting collaborative business processes between players in different enterprises (Chen, page 1, paragraph 1). Particularly, Chen describes that a collaborative business process with a plurality of work nodes, and explains that messages are used to synchronize between peer processes and to exchange data:

[0023] According to one embodiment, an inter-enterprise collaborative process management method and system are provided. A collaborative business process for modeling inter-enterprise collaboration (e.g., peer-to-peer (P2P) or business-to-business (B2B) interaction) is defined. The collaborative business process involves at least two players from different enterprises. The collaborative business process has a plurality of work nodes or tasks. Each work node has a task-role identifier for identifying a particular player that is responsible for executing each work node. A first collaborative process manager (FCPM) associated with the first player is provided to execute a first peer instance of the collaborative business process. A second collaborative process manager (SCPM) associated with the second player is provided to execute a second peer instance of the collaborative business process. Messages are employed for synchronization of the first and second peer process instances and for the exchange of data therebetween. (Chen, page 2, paragraph 23)

But nowhere does Chen describe that the messages’ headers are defined in accordance with a message class of the messages that is determinative of the content and configuration of the messages’ headers. Indeed, Chen does not even discuss headers of any type or configuration. Chen

certainly does not describe message headers defined in accordance with such message classes that include the application-message class, the application-response class, the application-error class and/or the system-acknowledge class recited in applicant's claim 1. Accordingly, Chen too fails to disclose or suggest at least the features of "defining an application message having a structured application message header, the structured message header being defined in accordance with a message class of the application message determinative of content and configuration of the message header," required by applicant's independent claim 1 or "the message class having one of multiple possible values including: a first value representative of an application-message class associated with application messages that cause specified operations to be performed at the receiving application, a second value representative of an application-response class associated with messages responsive to the application messages of the application-message class, a third value representative of an application-error class associated with error messages indicative of errors occurring at the receiving application processing the application messages and a fourth value representative of a system-acknowledge class associated with acknowledgement messages indicative that one or more application messages have been received by the receiving application," required by applicant's independent claim 1.

Because neither Traversat nor Chen discloses or suggests, alone or in combination, at least the features "defining an application message having a structured application message header, the structured message header being defined in accordance with a message class of the application message determinative of content and configuration of the message header," required by applicant's independent claim 1 or "the message class having one of multiple possible values including: a first value representative of an application-message class associated with application messages that cause specified operations to be performed at the receiving application, a second value

representative of an application-response class associated with messages responsive to the application messages of the application-message class, a third value representative of an application-error class associated with error messages indicative of errors occurring at the receiving application processing the application messages and a fourth value representative of a system-acknowledge class associated with acknowledgement messages indicative that one or more application messages have been received by the receiving application,” applicant’s independent claim 1 and the claims depending from it are patentable over the cited art.

Additionally, applicant further contends that, contrary to the examiner’s contentions, the cited references also fail to disclose or suggest at least the feature “defining an application message having a structured application message header, the structured message header being defined … in accordance with a messaging protocol of a business application of a collaborative business enterprise.”

Specifically, in rejecting claim 1, admitted that “Traversat does not disclose a business application to manage and control a business enterprise or a processing mode for the message” (Office action, page 6). Traversat also fails to disclose or suggest at least the features of “defining an application message having a structured application message header, the structured message header being defined … in accordance with a messaging protocol of a business application of a collaborative business enterprise.” The examiner, however, relies on Chen as allegedly disclosing this feature, and states, “[h]owever, Chen discloses a business application to manage and control a business enterprise (page 3; [0047], lines 1-12)” (Office action, page 6). Applicant disagrees.

As noted above, Chen describes a peer-to-peer collaborative process management method and system for supporting collaborative business processes between players in different enterprises. Chen explains that a collaborative process is based on an operational protocol (e.g., an on-line

purchase protocol or an auction protocol) that defines which party in a collaborative network (involving multiple parties operating from different nodes) performs which operation.

[0047] As described previously, the collaborative process management mechanism of the present invention involves a collaborative business process. A collaborative business process involves multiple parties (also referred to herein as players or partners). The collaborative business process definition is based on a commonly agreed operational protocol, such as the protocol for on-line purchase or the protocol for an auction. In contrast to prior art approaches, the collaborative business process is not executed by a centralized workflow engine, but the present invention employs multiple engines (e.g., FCPM 54 and SCPM 58) to collaboratively execute the collaborative business process. (Chen, page 3, paragraph 47)

Chen, however, does not describe a protocol (e.g., a business application protocol) according to which message headers are defined. Indeed, Chen does not even discuss message header. Chen collaborative operational protocols are not the same as a messaging protocol to define a structured messaging header. Accordingly, Chen fails to disclose or suggest at least the features of “defining an application message having a structured application message header, the structured message header being defined … in accordance with a messaging protocol of a business application of a collaborative business enterprise,” as required by applicant’s independent claim 1.

Because neither Traversat nor Chen discloses or suggests, alone or in combination, at least the features “defining an application message having a structured application message header, the structured message header being defined … in accordance with a messaging protocol of a business application of a collaborative business enterprise,” for this reason too applicant’s independent claim 1 and the claims depending from it are patentable over the cited art.

Furthermore, with respect to claim 1, applicant further contends that none of references discloses or suggests the features of “at least one of the one or more components of the structured header including information related to: a processing mode for the message, the processing mode having one of a multiple of values indicative of whether a reply responsive to the application message is to be transmitted to the sending application upon processing of the application message

by the receiving application.” Accordingly, for this reason too applicant contends that applicant’s independent claim 1 and the claims depending from it are patentable over the cited art.

Applicant’s independent claims 5, 9, 10 and 11 recite “defining an application message having a structured application message header, the structured message header being defined in accordance with a message class of the application message determinative of content and configuration of the message header and in accordance with a messaging protocol of a business application of a collaborative business enterprise,” and “wherein the message class having one of multiple possible values including: a first value representative of an application-message class associated with application messages that cause specified operations to be performed at the receiving application, a second value representative of an application-response class associated with messages responsive to the application messages of the application-message class, a third value representative of an application-error class associated with error messages indicative of errors occurring at the receiving application processing the application messages and a fourth value representative of a system-acknowledge class associated with acknowledgement messages indicative that one or more application messages have been received by the receiving application.” For reasons similar to those provided with respect to independent claim 1, at least these features are not disclosed by the cited art. Applicant’s independent claims 5, 9, 10 and 11, and the respective claims depending from them, are therefore patentable over the cited art.

CONCLUSION

It is believed that all of the pending claims have been addressed in this paper. However, failure to address a specific rejection, issue or comment, does not signify agreement with or concession of that rejection, issue or comment. In addition, because the arguments made above are not intended to be exhaustive, there may be reasons for patentability of any or all pending claims (or

other claims) that have not been expressed. Finally, nothing in this paper should be construed as an intent to concede any issue with regard to any claim, except as specifically stated in this paper, and the amendment of any claim does not necessarily signify concession of unpatentability of the claim prior to its amendment.

On the basis of the foregoing amendments, applicant respectfully submits that the pending claims are in condition for allowance. If there are any questions regarding these amendments and remarks, the examiner is encouraged to contact the undersigned at the telephone number provided below.

Along with this Response to Final Office Action, Applicant hereby submits a Request for Continued Examination along with a Petition for an Extension of Time.

The Commissioner is hereby authorized to charge any fees that may be due, or credit any overpayment of same, to Deposit Account No. 50-0311, Reference No. 34874-096.

Respectfully submitted,

Ido Rabinovitch
Ido Rabinovitch
Reg. No. L0080

Date: February 13, 2009

MINTZ, LEVIN, COHN, FERRIS
GLOVSKY and POPEO, P.C.
Attorneys for Applicant
One Financial Center
Boston, Massachusetts 02111
Customer No. 64280
Telephone: 617/348-1806
Facsimile: 617/542-2241
email: irabinovitch@mintz.com